ELBUG
SUPPLEMENT TO BEEBUG

VOLUME 3 NUMBER 4
MARCH 1986

News News News News News News Ne

The RS423 interface being developed by Acorn has finally seen the light of day under the Pace label. Acorn approached Pace with a nearly finished interface to plug into the Plus-1 and Pace has finished off the hardware and packaged it in a complete comms pack. For £160 the pack provides the RS423 interface, Pace's popular Nightingale modem (see BEEBUG review in Vol.3 No.9), a special Electron version of the Commstar communications ROM, and a free quarter's subscription to Micronet and Microlink on Telecom gold. Pace is on 0274-488211.

High Scores

The following are the latest high scores by ELBUG members. We are always pleased to display the results of your hours toiling at the fire key, so keep them coming.

Supplier	Game	Score	Player
ACORNSOFT	Freefall	997	A.Lee
ADDICTIVE	Boffin	230405	M.Lancaster
ELBUG MAG	Digger	3100	M.Jones
ELBUG MAG	Flowers of Hell	4035	T.Jones
ELBUG MAG	Spider Man	1590	I.G.Lee
IJK	Hyperdrive	440	M.Jones
QUICKSILVA	Mined Out	7380	M.Jones
ROMIK	Birds of Prey	55310	M.Jones
SOFT INVASION	Blitzkrieg	64000	M.Jones

G CASSETTE ELBUG CASSETTE ELBUG CASSETTE ELBUG CASSETTE ELBUG CASSETTE ELBUG CASSETTE E

This month's ELBUG magazine cassette includes several of the programs from this month's BEEBUG - the windows and icons utilities and demonstration, the First Course examples on EVAL, and the Burger Time game - plus the five programs from the PLOT 77 article and the Sliding Block Puzzle game in this month's ELBUG supplement. All this for just £3.00 plus 50p post and packing. ELBUG cassettes from Vol.3 No.3 onwards are available to order only. Please address all orders to the St. Albans address, mark the envelope 'ELBUG cassette', and allow at least ten days for delivery.

Your Guide to BEEBUG

No alterations are required to the programs appropriate to the Electron from this month's BEEBUG. In fact, most of the programs in BEEBUG this month will work on an Electron. A few general comments on converting Beeb programs to the Electron may also prove useful.

Beeb programs sometimes fail to run properly on the Electron because of the Beeb's mode 7 and extra sound facilities. However, conversion rarely proves very difficult.

The Beeb's mode 7 display is of the same format as the mode 6 display that ELBUG members know but with a full range of colours and crude graphics. Mode 7 is not available on the Electron. However, the Electron defaults to mode 6 when it encounters a MODE 7 command. So BBC micro programs that include a mode 7 display will therefore run on the Electron but often with a corrupted (and monochrome) display.

The Beeb has three sound channels each with full control over the volume of notes. Electron Basic includes all the SOUND and ENVELOPE parameters that are used in the BBC micro even though several have no effect. This means that Beeb programs making full use of sound will still run on the Electron unaltered, though not, of course, producing the full sound effect.

The other major difference between the Electron and the BBC micro is speed. The Electron is much slower, and particularly slow in modes 0, 1, and 2. BBC micro programs often benefit from a change of mode (admittedly, with a loss of some colours) when running them on the Electron. When changes are needed they will be given in this ELBUG supplement.

THE MASTER 128 IN PRACTICE

Any ELBUG member tiring of the restrictions of the Electron may like to look to the Master 128 as an upgrade path. What could be better: the same manufacturer, the same dialect of Basic and BEEBUG there to support you.

WEATHER SATELLITES

Although the two systems looked at here are not able to be connected to the Electron, ELBUG members may still find much to interest them in this fascinating field.

AMXTRAS

Yet more goodies from the AMX stables. Unfortunately there is still no version of the AMX mouse available for the Electron.

WINDOWS AND ICONS

If you are disappointed at missing out on the mouse boom then this excellent set of utilities will give your computer the user-friendliness of the AMX software without even requiring a mouse.

All that you need for professional-looking windows and icon menus is here for inclusion in your own programs; and it all works on the Electron without modification.

PRESERVING SCREENS

This handy utility to swap between screen displays won't work on the Electron for two reasons.

Firstly, for useful application one of the screens should really be a mode 7 screen for memory economy (the Electron, of course does not have a mode 7 display). The second reason for the program not working is much more important. The program directly accesses the registers of the display chip used in the Beeb. The Electron does not have this chip but incorporates all its functions in the ULA so the program cannot be made to work on the Elk.

RAMSAFE

This fascinating utility will make regular backup copies of your program, as you develop it, without ever using a cassette or disc. The backup is made to sideways RAM.

ELBUG members with a sideways RAM adaptor of some kind (such as the ACP RAM module reviewed in ELBUG Vol.2 No.10 or the Slogger ROM board reviewed in ELBUG

Vol.1 No.10 equipped with static RAM) may be able to make use of this program but some alterations (depending on the SWR system) will have to be made. Not for the faint-hearted!

ADVENTURE GAMES

Two of the adventure games reviewed this month are compatible with the Electron. These are 'Terrormolinos' and 'Hampstead'.

GETTING A BETTER VIEW

The excellent View word processor is, of course, available for the Electron as a cartridge for the Plus-1. All users of View would do well to see how this powerful software can be fully exploited. Even if you do not own a copy of View at present, this article will show you some of the features that the word processor can provide.

ANGELS AT 12 O'CLOCK

There's nothing better than taking your micro up for a couple of quick circuits around the airfield. The software reviewed here allows you to do just this.

Two of the three flight simulators reviewed in this article have Electron compatible versions for you to try out. These are 'Strike Force Harrier' from Mirrorsoft and 'Jump Jet' from Anirog.

GAMES REVIEWS

Only one of the games reviewed this month is available in an Electron version. This is the excellent 'Citadel' from Superior Software.

WORKSHOP

The 6522 Timers

Unfortunately the Electron has no user VIA (6522) to provide timers, so this article will be little use to most ELBUG members.

However, if you have a User Port interface which contains a 6522 chip (such as the Mushroom Printer and User Port interface reviewed in Elbug Vol.2 No.4) then you will be able to make use of the routines given here. Note, however, that the addresses of the VIA registers given in the article are wrong for such an interface (for the Mushroom interface, for example, you should subtract &100 from the addresses given) and you should consult the interface manual.

THREE MODEMS

None of the modems reviewed here are directly connectable to the Electron because they all require a serial interface. However, ELBUG members eager to enter the world of comms will be pleased to see the comms package for the Electron available from Pace (see front cover of this supplement).

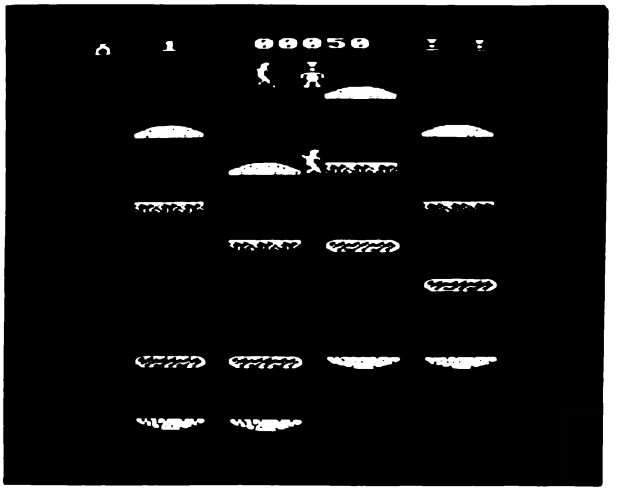
WRITING YOUR OWN COMPILER

Although the program that accompanies this series will not work on the Electron because of the differences between Basic in the Beeb and the Elk, ELBUG members may be interested to see how a compiler is written.

FIRST COURSE

Using the EVAL function (Part 2)

Mike Williams continues his introduction to the useful EVAL function in this second article. All the remarks on this function and the programs that accompany the article are equally applicable to the Electron.



BURGER TIME

This fast, furious, and not a little crazy, game puts you in charge of cooking the hamburgers in the kitchen of that well known chain of fast food shops, MacWimpking.

The game works very well on the Electron. The title and high score screens use mode 7 so these will appear rather corrupted on your machine. However, the main action works fine.

PLOT 77 EXAMINED

Gordon Wootton delves into the inner workings of the PLOT 77 command and finds that it is a very useful feature on the Electron but one that is sadly under-used.

The normal method used on the Electron to fill in an area of the screen with a colour is based on the PLOT 85 command (see the User Guide, page 176). However, this method is not so useful when an irregular shape or one that is made up of large number of straight lines, or even one containing a 'hole' is to be plotted.

In these cases a solution can often be found in the use of the PLOT77 command. This is briefly described in the User Guide on page 99 but, strangely, programmers do not normally make the most of the potential of this routine.

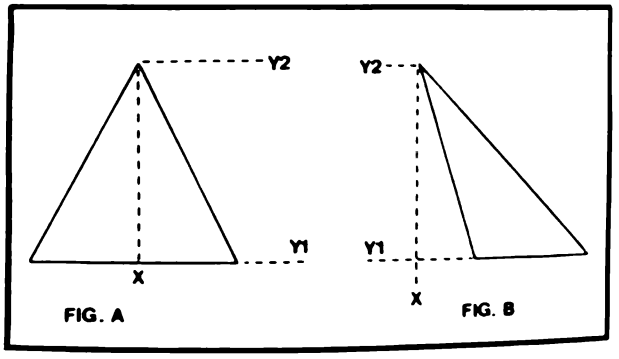
PLOT 77 (and its associated other PLOT codes used for relative plotting, colour inverting, and so on) performs a very simple function. Issuing the command PLOT77,X,Y will cause the Electron to go to the point, X,Y on the screen and search either side of this point until it finds a pixel that is lit with any other colour than the background colour. When the first pixels are found on either side of the specified point then a line is drawn between them in the current foreground colour.

By repeating this process several times, incrementing the value of Y each time, an area of the screen, bounded by colour that's not the background colour, is filled in with the foreground colour. This means that an outline of a shape (drawn with the normal DRAW command) can be simply filled in by repeated PLOT 77 commands plotted up its whole length. The first program illustrates this.

```
10 REM PROGRAM triangle
20 MODE 1
30 GCOL0,2
40 S%=4
50 MOVE 200,200:DRAW 900,200
60 DRAW 700,800:DRAW 200,200
70 FOR Y%=200 TO 800 STEP S%
80 PLOT77,700,Y%
90 NEXT Y%
100 END
```

The increment (S%) in the Y co-ordinate, as the PLOT 77 commands are repeated, is set in line 40 to 4. This is because the vertical scale of the Electron's graphics co-ordinate system is 1024 whereas there are actually only 256 pixels displayed vertically on the screen. Decreasing the step size will still fill the triangle but the process takes longer as each horizontal line of pixels is checked and filled more than once. Increasing the value of S% will give you a striped triangle as some lines of pixels are missed out altogether. Try altering line 40 to use different values of S%.

So, then, in its simplest form, the fill procedure needs two Y co-ordinates (Y1 and Y2) and one X co-ordinate. These are used in a FOR-NEXT loop to give successive PLOT 77 commands up the shape.



One limitation of the PLOT77 routine is illustrated in the diagram. Figure A is a situation as we have just seen and the fill would be successful. In figure B, however, the X co-ordinate corresponding to the maximum difference of Y co-ordinates lies outside the shape to be filled. Such shapes can be filled with PLOT77, but only by a more complex method: either the X co-ordinate must be changed as the loop progresses or more than one

loop with different X co-ordinates must be used.

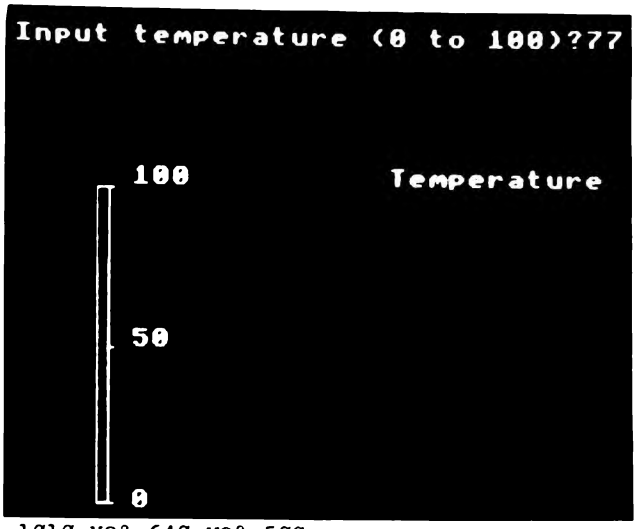
Clearly, PLOT 77 offers no advantage over PLOT 85 for the filling of such simple shapes as a triangle but the smooth fill of the distorted H in the second program shows its usefulness. Here, two plot statements are required. Try leaving one out to see what happens.

```
  10 REM PROGRAM letter
  20 MODE 1
  30 GCOL0,2
  40 MOVE 130,900:DRAW 180,900
  50 DRAW 180,600:DRAW 700,600
  60 DRAW 700,900:DRAW 900,900
  70 DRAW 900,500:DRAW 870,500
  80 DRAW 870,50:DRAW 750,50
  90 DRAW 750,500:DRAW 300,500
 100 DRAW 300,50:DRAW 100,50
 110 DRAW 100,500:DRAW 130,500
 120 DRAW 130,900
 130 FOR Y%=50 TO 900 STEP 4
 140 PLOT77,150,Y%
 150 PLOT77,800,Y%
 160 NEXT Y%
 170 END
```

As the PLOT 77 command is concerned only with an individual horizontal line of a shape at a time, a particularly useful feature of this command is the ability to change colour within the fill or to part-fill an area. The following program uses this idea to produce a circle filled with stripes.

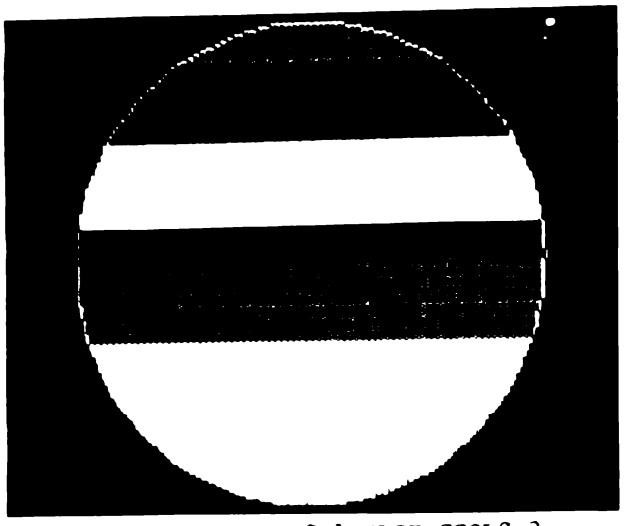
The somewhat complex circle drawing routine is used as it produces accurate circles very fast. You can simply change the colour and position of the stripes by altering the parameters in the calls to the procedure PROCstripe in lines 50 to 90. The first two figures are the bottom and top co-ordinates of the stripe and the third the colour.

```
  10 REM PROGRAM striped circle
  20 MODE 1
  30 GCOL0,2
  40 PROCdraw
  50 PROCstripe(200,400,2)
  60 PROCstripe(400,500,1)
  70 PROCstripe(550,600,3)
  80 PROCstripe(600,650,2)
  90 PROCstripe(750,799,1)
 100 END
 110 :
1000 DEFPROCdraw
1010 XO%=640:YO%=500
1020 N%=100:T=2*PI/N%
1030 S=SIN(T):C=COS(T)
1040 R%=300:X1=R%:Y1=0
1050 MOVE XO%+X1,YO%+Y1
1060 FOR M%=1 TO N%
1070 X=X1*C-Y1*S
1080 Y=X1*S+Y1*C
1090 DRAW XO%+X,YO%+Y
1100 X1=X:Y1=Y
1110 NEXT M%
1120 ENDPROC
1130 :
1140 DEFPROCstripe(B%,T%,C%)
1150 GCOL0,C%
1160 FOR Y%=B% TO T% STEP 4
1170 PLOT77,XO%,Y%
1180 NEXT Y%
1190 ENDPROC
```



This part-filling of a shape can be put to advantage in many ways. A simple example is shown in the thermometer program below. This represents a number graphically by the part-filling of a scaled column.

```
  10 REM PROGRAM thermometer
  20 MODE 1
  30 REPEAT
  40 CLS
  50 INPUT TAB(2,2)"Input temperature (
0 to 100)";temp%
  60 IFtemp%<0 OR temp%>100 PRINT ''"
out of range":GOTO 80
  70 PROCfill(temp%,(temp%>35))
  80 I=INKEY(100)
  90 UNTIL FALSE
 100 END
 110 :
1000 DEFPROCfill(L%,C%)
1010 PROCdraw
```



```
1020 IF C% THEN GCOL0,1 ELSE GCOL0,3
1030 FOR Y%=200 TO 200+5*L% STEP 4
1040 PLOT77,210,Y%
1050 NEXT Y%
1060 ENDPROC
1070 :
1080 DEFPROCdraw
1090 GCOL0,2
1100 MOVE 230,700:DRAW 200,700
1110 DRAW 200,200:DRAW 230,200
1120 MOVE 220,200:DRAW 220,700
1130 MOVE 220,450:DRAW 230,450
1140 PRINTTAB(1,9)100;TAB(20)"Temperatu
re";TAB(8,17)50;TAB(8,25)0
1150 ENDPROC
```

The fast circle routine is also used in the next program which demonstrates the ease of filling shapes within shapes. The program draws three squares with circles inside them and fills each one with a different colour. Flashing colours could also be used if preferred. Although the program is quite complicated, you can imagine how much longer it would be if the other PLOT commands were used.

```
  10 REM PROGRAM fillup
  20 MODE 2
  30 XO%=640:YO%=500
  40 width1=150:PROCpattern(width1)
  50 width2=450:PROCpattern(width2)
  60 width3=800:PROCpattern(width3)
  70 PROCfill1(width1,0,1)
  80 PROCfill1(width2,width1/2,2)
  90 PROCfill1(width3,width2/2,3)
 100 PROCfill2(width1/2,4)
 110 PROCfill2(width2/2,5)
 120 PROCfill2(width3/2,6)
 130 END
 140 :
1000 DEFPROCcircle(R)
1010 GCOL0,1
1020 N%=100:T=2*PI/N%
1030 S=SIN(T):C=COS(T)
1040 X1=R:Y1=0
1050 MOVE XO%+X1,YO%+Y1
1060 FOR M%=1 TO N%
1070 X=X1*C-Y1*S
1080 Y=X1*S+Y1*C
1090 DRAW XO%+X,YO%+Y
1100 X1=X:Y1=Y
1110 NEXT M%
1120 ENDPROC
1130 :
1140 DEFPROCpattern(width%)
1150 GCOL0,7
1160 W%=width%/2
1170 MOVE XO%-W%,YO%-W%:DRAW XO%-W%,YO%
+W%
1180 DRAW XO%+W%,YO%+W%:DRAW XO%+W%,YO%
-W%:DRAW XO%-W%,YO%-W%
1190 PROCcircle(width%/2-30)
1200 ENDPROC
1210 :
1220 DEFPROCfill1(width,limit,C%)
1230 radius=width/2-30
1240 GCOL0,C%
1250 FOR N%=limit TO radius STEP 4
1260 PLOT77,XO%,YO%+N%
1270 PLOT77,XO%,YO%-N%
1280 NEXT N%
1290 FOR Y%=YO%-limit TO YO%+limit+4 ST
EP 4
1300 PLOT77,XO%+limit+10,Y%
1310 PLOT77,XO%-limit-10,Y%
1320 NEXT Y%
1330 ENDPROC
1340 :
1350 DEFPROCfill2(width,C%)
1360 GCOL0,C%
1370 FOR Y%=YO%-width TO YO%+width STEP
 4
1380 PLOT77,XO%-width+10,Y%
1390 PLOT77,XO%+width-10,Y%
1400 NEXT Y%
1410 ENDPROC
```

PROCfill1 is used to fill the circles around the boxes. This is done in two parts: the two hemispheres on top and below the inner square are filled first and then the sections on either side of the box. PROCfill2 fills in the boxes around the circles. This is done simply in one go by a succession of PLOT 77 commands up the two sides of the box.

Using the PLOT 77 in this way can greatly simplify your programs. Complex shapes can be easily filled in with colour using far less code and faster than you could achieve otherwise. ●

SLIDING BLOCK PUZZLE

You've probably seen sliding block puzzles before. They usually fall out of the cheapest Christmas crackers. This version from John Crombie looks more like it has fallen out of an Apple Macintosh.

This is a computer version of the classic, but challenging, hand-held game. A grid of four by four blocks, with one block missing, is presented on the screen with very Macintosh shadows and background. Each block has a letter on it and the blocks are initially arranged in alphabetical order. They are then jumbled by the computer and it is your task to rearrange them into order again.

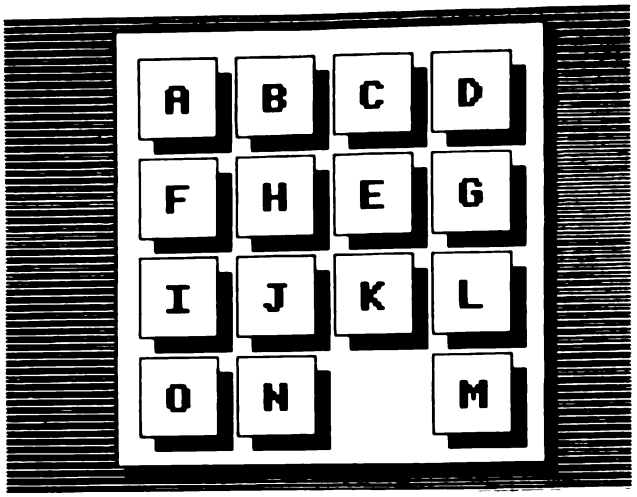
This is done by sliding one of the adjacent blocks into the empty space. With careful planning, and a bit of luck, you can successfully shuffle the blocks back into position whereupon the ever-helpful Electron will shuffle them all again!

The blocks are moved using the cursor control keys. So, to move the block below the empty space up into the space use the upwards cursor control key.

The program contains a section of machine code (PROCassemble) to speed things up a bit so it is recommended that when you have typed it in you save the program before running it, in case of any disastrous errors.

Like all the best games the sliding block puzzle is very simple but that doesn't make it any easier. You will find this very pretty representation addictive in the extreme.

```
   10 REM PROGRAM SLIDING
   20 REM VERSION E0.2
   30 REM AUTHOR  JOHN CROMBIE
   40 REM ELBUG   MARCH 1986
   50 REM PROGRAM SUBJECT TO COPYRIGHT
   60 :
  100 MODE4:VDU5
  110 shdw%=20:DIM letter 255,GRID(15)
  120 oswrch=&FFEE:osword=&FFF1:X%=&70:Y
%=0:A%=10:byte=&71:mem=&7A:userch=&7B
  130 PROCbackground
  140 PROCAssemble
  150 PROCdefine
  160 VDU19,0,7;0;19,1,0;0;5
  170 PROCsquare(250,70,876)
  180 FOR K%=0 TO 15
  190 PROCput(K%,K%)
  200 NEXT K%
  210 PROCmix
  220 REPEAT
  230 IF INKEY(-26) THEN N%=space%+1:IF
N%>15 THEN N%=space%
  240 IF INKEY(-122) THEN N%=space%-1:IF
 N%<0  THEN N%=space%
  250 IF INKEY(-58) THEN N%=space%+4:IF
N%>15 THEN N%=space%
  260 IF INKEY(-42) THEN N%=space%-4:IF
N%<0  THEN N%=space%
  270  IF INKEY(-102) THEN PROCmix
  280  IF N%=space% THEN 230
  290 PROCmove
  300 finish%=1
  310 FOR I%=0 TO 15
  320 IF GRID(I%)<>I% THEN finish%=0
  330 NEXT I%
  340 UNTIL finish%=1
  350 PROCfinish:RUN
  360 END
  370 :
 1000 DEF PROCcsquare(X,Y)
 1010 GCOL0,1
 1020 MOVEX-32,Y+180:PRINTS$
 1030 ENDPROC
 1040 DEF PROCsquare(X,Y,size%)
 1050 GCOL0,1:PROCsqu(X,Y,size%)
 1060 GCOL0,0:PROCsqu(X-shdw%,Y+shdw%,si
ze%)
 1070 GCOL0,1
 1080 MOVE X+size%-shdw%,Y+size%+shdw%:P
LOT1,-size%,0:PLOT1,0,-size%
 1090 PLOT1,size%,0:PLOT1,0,size%
 1100 ENDPROC
 1110 DEF PROCsqu(X%,Y%,size%)
 1120 MOVEX%,Y%:PLOT0,0,size%
 1130 PLOT81,size%,-size%
 1140 PLOT81,0,size%
 1150 ENDPROC
 1160 DEF PROCbackground
 1170 FOR I%=0 TO 1020 STEP 8
 1180 VDU18,0,135,24,0;I%;1279;I%;12
 1190 NEXT I%:VDU18,0,128,26
 1200 ENDPROC
 1210 DEF PROCAssemble
 1220 FOR I%=0 TO 2 STEP 2
 1230 P%=letter
 1240 [OPT I%
 1250 JSR osword
```




```
1260 LDA #224:STA userch
1270 JSR vdu23:JSR definel
1280 INC userch:JSR vdu23:JSR definel
1290 INC userch:JSR vdu23:JSR define2
1300 INC userch:JSR vdu23:JSR define2
1310 RTS
1320 .vdu23
1330 LDA #23:JSR oswrch
1340 LDA userch:JSR oswrch
1350 RTS
1360 .definel:LDY #0
1370 .nextbyte LDA byte,Y:STA mem
1380 JSR expand
1390 INY:CPY#4:BNE nextbyte
1400 RTS
1410 .define2:LDY #4
1420 .nextbytel LDA byte,Y
1430 STA mem
1440 JSR expand
1450 INY:CPY #8:BNE nextbytel
1460 RTS
1470 .expand LDX #4
1480 .back ROL mem
1490 PHP:ROLA:PLP:ROLA
1500 DEX:BNE back
1510 JSR oswrch
1520 JSR oswrch
1530 LDA mem:STAbyte,Y
1540 RTS
1550 ]:NEXT
1560 char$=CHR$(224)+CHR$(225)+CHR$(10)
+CHR$(8)+CHR$(8)+CHR$(226)+CHR$(227)
1570 ENDPROC
1580 DEF PROCdefine
1590 VDU23,230,&80FF;&8080;&8080;&8080;
1600 VDU23,231,&FF;0;0;0;
1610 VDU23,232,&1FF;&101;&101;&101;
1620 VDU23,233,&8080;&8080;&8080;&8080;
1630 VDU23,234,&8080;&8080;&8080;&FF80;
1640 VDU23,235,&FFFF;&FFFF;&FFFF;&FFFF;
1650 B$=CHR$(10)+CHR$(8)+CHR$(8)+CHR$(8
)+CHR$(8)+CHR$(8)+CHR$(8)
1660 A$=CHR$(233)+"     "+CHR$(235)
1670 S$=CHR$(230)+CHR$(231)+CHR$(231)+C
HR$(231)+CHR$(231)+CHR$(231)+CHR$(233)+B$+A$+B$+A$
+B$+A$+B$+A$+B$+CHR$(231)+CHR$(235)+CHR$
(235)+CHR$(235)+CHR$(235)+CHR$(235)
1680 ENDPROC
1690 DEF PROCput(loc%,sym%)
1700 I%=loc% MOD 4
1710 J%=loc% DIV 4
1720 IF sym%=15 THEN PROCblank:ENDPROC
1730 PROCcsquare(I%*200+310,(3-J%)*200+130)
1740 ?X%=sym%+65
1750 CALL letter
1760 MOVE I%*200+324,(3-J%)*200+252
1770 PRINTchar$
1780 SOUND0,-10,4,1
1790 ENDPROC
1800 DEF PROCmix
1810 FOR M%=0 TO 15:GRID(M%)=M%:NEXT M%
1820 b%=3:c%=3:OS%=0:space%=15
1830 FOR M%=1 TO 100
1840 REPEAT
1850 S%=RND(4)
1860 UNTIL S%+OS%<>5
1870 IF S%=1 N%=space%-4
1880 IF S%=2 N%=space%-1
1890 IF S%=3 N%=space%+1
1900 IF S%=4 N%=space%+4
1910 IF N%<0 OR N%>15 THEN 1940
1920 OS%=S%
1930 PROCmove
1940 NEXT M%
1950 N%=space%
1960 ENDPROC
1970 DEF PROCblank
1980 GCOL0,0:PROCsqu(I%*200+270,(3-J%)*
200+120,196)
1990 space%=loc%
2000 b%=I%:c%=J%
2010 ENDPROC
2020 DEF PROCmove
2030 IF N%=space% THEN ENDPROC
2040 B%=N% MOD 4
2050 C%=N% DIV 4
2060 IF B%<>b% AND C%<>c% THEN ENDPROC
2070 IF ABS(B%-b%)>1 OR ABS(C%-c%)>1 TH
EN ENDPROC
2080 IF N%<0 OR N%>15 THEN ENDPROC
2090 GRID(space%)=GRID(N%)
2100 GRID(N%)=15
2110 PROCput(space%,GRID(space%))
2120 PROCput(N%,15)
2130 ENDPROC
2140 DEF PROCfinish
2150 VDU19,0,0;0;19,1,7;0;
2160 FOR I%=0 TO 15
2170 SOUND 1,-15,60+10*(I% MOD 3),1
2180 NEXT I%
2190 I=INKEY(100)
2200 VDU19,0,7;0;19,1,0;0;
2210 ENDPROC
```